// INFOWOK · AI ENGINEERING

The AI Engineering Roadmap

InfoWok — for senior software engineers · www.infowok.com

The exact path from Python basics to a deployed, observable AI agent. No course-selling, no fluff.

Every roadmap tells you to spend six months on linear algebra before you are "allowed" to build an AI agent. That advice is exactly why most people quit. This agentic AI roadmap flips the order: you build something that works in week one, then backfill the theory only when a real project needs it.

I learned agents this way myself, coming from a Python and Java background, and the project-first path kept me going where the math-first one had stalled me twice. By the end of this guide you will know whether agentic AI is worth your time in 2026, the exact order to learn things, what to skip, and the first three projects to put in your portfolio.

A quick definition before we start. **An agentic AI system is an LLM that can plan, call tools, and act in a loop toward a goal** — not just answer one prompt. If that idea is still fuzzy, read what AI agents actually are first, then come back here for the path.

Key takeaways

- **Build first, backfill theory later.** Skip the six-month math detour — you need solid Python and the ability to call an API, not linear algebra.
- **The path is five ordered stages:** Python → LLM basics → your first agent (the ReAct loop) → tools, MCP, and memory → deploy and build a portfolio.
- **Learn one framework deeply.** Start with CrewAI for a fast win, graduate to LangGraph for production; the concepts transfer, so the second framework takes days.
- **A working portfolio beats certificates.** Ship three projects — a research agent, a workflow bot, and a small multi-agent system.

Is Agentic AI Worth Learning in 2026?

Here is the honest answer: yes, but for builders, not for certificate collectors. 2026 is the year agents crossed from demos into production. Gartner expects 40% of enterprise applications to ship task-specific AI agents by the end of the year, up from under 5% a year earlier. That shift created real, paid demand for people who can build and run agents.

The money follows the demand, especially in India. Market data puts agentic AI engineer salaries roughly in the **₹8–30+ LPA** range, with senior, specialised roles going higher. LLM engineering and agentic AI carry some of the biggest skill premiums in the market, and AI-first product companies often pay two to three times what IT-services firms pay for the same title. India is also one of the largest markets searching for and hiring these skills right now.

But I will not oversell it. Salary figures are market signals, not promises, and they move fast. The field is new enough that most teams are upskilling existing engineers into agent roles rather than hiring fresh, so **a working portfolio matters far more than another course certificate.** That is good news: it means you can earn your way in by building, which is exactly what this roadmap optimises for.

The Agentic AI Roadmap, Step by Step

This is the path I would give any developer starting today. Treat it as a checklist you can save and return to — finish each stage before the next.

The diagram shows the whole route. Here is each stage in plain terms:

- **Stage 1 — Python fundamentals.** Functions, data structures, working with REST APIs, and a little async. You do not need advanced Python, but you must be comfortable reading and debugging it. If you are still new to coding, my note on going from AI hype to hands-on is a gentler on-ramp.
- **Stage 2 — LLM basics.** How tokens, prompts, context windows, and model APIs work. Spend a week calling a model directly so you understand inputs and outputs before any framework hides them.
- **Stage 3 — Your first agent.** Build a single agent using the ReAct loop (Reasoning + Acting), the most beginner-friendly pattern in 2026 because you can watch every decision the agent makes.
- **Stage 4 — Tools, MCP, and memory.** Give the agent real capabilities: external tools, a standard connection layer via MCP, and short- and long-term memory so it remembers past turns.
- **Stage 5 — Deploy and build a portfolio.** Ship the agent to a public URL and write up what it does. A live demo beats ten finished tutorials.

The order is the point. Each stage gives you something runnable, so motivation compounds instead of draining away.

What to Skip at the Start (and Learn Later)

The fastest way to quit is to front-load theory you do not need yet. So skip these three things early.

First, **skip the six-month math detour.** You do not need linear algebra or calculus to build agents in 2026 — you need to call models and wire up tools. Pick up the math later if you move into research or fine-tuning.

Second, skip training models from scratch. You will use hosted models through an API. Learning to train one is a separate career track, and it will not help you ship your first agent.

Third, skip learning every framework at once. I wasted a weekend hopping between three of them before realising the concepts are the same underneath. **Learn one framework deeply, and the rest take days.** Build first; backfill theory only when a real project blocks you.

The Tools You'll Actually Touch: LangGraph vs CrewAI vs Pydantic AI

You will hear a dozen tool names. In practice, three matter for most builders, and you should start with exactly one.

Here is how I would choose. **CrewAI** gets you a working multi-agent demo in two to four hours — it is the best confidence booster for a first project.

LangGraph became the 2026 production standard for stateful, auditable workflows, with the largest monthly download numbers of the orchestration frameworks (LangChain keeps a useful framework overview), so it is worth graduating to once you want control. **Pydantic AI** is the cleanest choice for a single, type-safe agent that returns structured output you can trust.

My advice: start with CrewAI for the quick win, then move to LangGraph as your projects get serious. For a deeper side-by-side, see my LangGraph vs CrewAI vs AutoGen comparison, and if type safety appeals to you, the Pydantic AI tutorial walks through a typed agent end to end.

Your First Three Portfolio Projects

Recruiters skim certificates and study demos. Build these three, in order, and you will have a portfolio that proves you can ship.

1. **A research agent.** Give it a question, let it search, read, and return a sourced summary. This teaches tool use and the ReAct loop on a problem everyone understands.
2. **A workflow automation bot.** Pick one annoying task — triaging emails, drafting replies, filing tickets — and automate it end to end. This shows you can connect an agent to real systems via tools and MCP.
3. **A small multi-agent system.** A supervisor that delegates to two or three worker agents. This is the capstone that proves you understand coordination, not just a single loop.

If you want a guided build for the deployment muscle, my agentic AI app series in Python takes one agent from zero to a deployed service.

Frequently Asked Questions

Is agentic AI just hype? The hype peaked in 2025; 2026 is the quieter, more useful phase where agents actually run in production. The skill is worth learning precisely because the novelty has worn off and the work is real.

Should I learn generative AI first? A little. Understand prompting and how an LLM responds, then move straight to agents. You do not need to master image or video generation to build agentic systems.

I only know basic Python — is that enough? Yes, to start. You will sharpen your Python as you build. Comfort with functions and APIs is the real prerequisite, not years of experience.

Conclusion

Agentic AI is worth learning in 2026 if you are willing to build, and this agentic AI roadmap is simpler than the gatekeepers make it sound: Python, LLM basics, a first agent, tools and memory, then deploy. Skip the math detour, pick one framework, and ship three projects.

Where are you on this roadmap right now — still on Python, or already wiring up your first agent? Tell me in the comments; I read them and point people to the next step.

Read next: Build an Agentic AI App in Python (Part 1). It turns Stage 3 of this roadmap into real, running code.